

## **Pemrograman Berorientasi Objek (OOP)**

Sabtu, 27 Juni 2015 | 15:58:36 WIB | **Endang Kurniawan**

OOP/PBO merupakan paradigma pemrograman yang populer saat ini yang telah menggantikan teknik pemrograman berbasis prosedur. Object Oriented Programming yang berarti pula Pemrograman Berorientasi Objek sudah ditemukan sekitar tahun 1960 dan dikembangkan pada permulaan tahun 1970.

Pemrograman Berorientasi Objek (Object Oriented Programming/OOP) merupakan pemrograman yang berorientasikan kepada objek, dimana semua data dan fungsi dibungkus dalam class-class atau object-object. Setiap object dapat menerima pesan, memproses data, mengirim, menyimpan dan memanipulasi data. Beberapa object berinteraksi dengan saling memberikan informasi satu terhadap yang lainnya.

Masing-masing object harus berisikan informasi mengenai dirinya sendiri dan dapat dihubungkan dengan Object yang lain. Pemrograman berorientasi objek berbeda dengan pemrograman prosedural yang hanya menggunakan satu halaman kebawah untuk mengerjakan banyak perintah atau statement. Penggunaan pemrograman berorientasi objek sangat banyak sekali, contoh : java, php, perl, c#, cobol, dan lainnya.

Dalam konsep Pemrograman Berorientasi Objek dikenal beberapa istilah umum, yaitu:

### **1. ATTRIBUTE**

#### ***Definisi dan Gambaran Umum Atribut***

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas, Atribut dapat memiliki hak akses private, public maupun protected.

Sebuah atribut yang dinyatakan sebagai private hanya dapat diakses secara langsung oleh kelas yang membungkusnya, sedangkan kelas lainnya tidak dapat mengakses atribut ini secara langsung.

Sebuah atribut yang dinyatakan sebagai public dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya.

Sebuah atribut yang dinyatakan sebagai protected tidak dapat diakses secara langsung oleh kelas lain di luar kelas yang membungkusnya, kecuali kelas yang mengaksesnya adalah kelas turunan dari kelas yang membungkusnya.

Atribut juga biasa disebut state/ properties/ field merupakan bagian yang dimiliki oleh sebuah class yang tidak melakukan operasi, tetapi kepadanya dilakukan operasi sehingga dapat merubah nilai dari attribute tersebut.

#### ***Karakteristik Atribut***

Dalam class, atribut disebut sebagai variabel. Atribut dapat membedakan antara satu object dengan object yang lain. Sebagai contoh, pada class : mahasiswa, terdapat object mahasiswa si A, dan object mahasiswa si B.

Yang membedakan antara object si A dan si B adalah NPM-nya (Nomor Pokok Mahasiswa) yang merupakan atribut dari object tersebut. Pada atribut, terdapat pula dua istilah variabel, yaitu Instance Variabel dan class variable. Dimana pada tiap variabel memiliki karakteristik tersendiri, diantaranya:

- Pada Instance Variabel tiap object memiliki instance variabel dan menyimpan nilainya tersendiri.
- Sedangkan pada Class Variabel atribut yang dimiliki oleh semua object yang berasal dari class yang sama, serta semua object memiliki nilai class variabel yang sama. atribut yang dimiliki object dari class yang sama.

## 2. METHOD

### *Definisi dan Gambaran Umum Method*

Method adalah fungsi atau prosedur yang dibuat oleh seorang programmer didalam suatu Class. Dengan kata lain, method pada sebuah kelas hampir sama dengan fungsi atau prosedur pada pemrograman prosedural.

Pada sebuah method di dalam sebuah kelas juga memiliki izin akses seperti halnya atribut pada kelas, izin akses itu antara lain private, public dan protected yang memiliki arti sama pada izin akses atribut yang telah dibahas sebelumnya. Sebuah kelas boleh memiliki lebih dari satu method dengan nama yang sama asalkan memiliki parameter masukan yang berbeda sehingga kompiler atau interpreter dapat mengenali method mana yang dipanggil.

Hal ini dinamakan overloading. Di dalam sebuah kelas, terdapat juga yang disebut sebagai method atau atribut statis yang memiliki kata kunci static. Maksud dari statis di sini adalah method yang dapat diakses secara berbagi dengan semua objek lain tanpa harus membuat objek yang memiliki method statis tadi (tanpa proses new), tapi sebuah method statis mempunyai keterbatasan yaitu hanya dapat mengakses atribut atau method lain di dalam kelas yang membungkusnya yang juga bersifat statis. Method statis biasanya diimplementasikan untuk method main.

### *Karakteristik Method*

Didalam suatu class, behavior disimpan dalam bentuk Method. Method merupakan serangkaian statemen / perintah (perintah = baris program) dalam suatu class yang menghandle task tertentu. Method merupakan hal-hal yang bisa dilakukan oleh object dari suatu class.

Method didefinisikan pada class akan tetapi dipanggil melalui object. Method memiliki peranan diantaranya :

- merubah nilai atribut dari suatu object,
- menerima informasi dari object lain, dan
- mengirim informasi ke obyek lain untuk melakukan suatu task (cara obyek berkomunikasi dengan obyek lain adalah dengan menggunakan method).

## 3. CLASS

Merupakan model yang berisi kumpulan attribute dan method dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh class manusia memiliki attribute berat, tinggi, usia kemudian memiliki method makan, minum, tidur. Method dalam sebuah class dapat merubah attribute yang dimiliki oleh class tersebut. Sebuah class merupakan dasar dari modularitas dan struktur dalam pemrograman berorientasi object.

### *Definisi dan Gambaran Umum Class*

Class didefinisikan sebagai sebuah blueprint(denah), atau prototipe, yang mendefinisikan variabel-variabel dan metode-metode yang umum untuk semua objek dari n jenis tertentu (n maksudnya jumlah tertentu). Sebuah kelas menyerupai sebuah struktur yang merupakan tipe data sendiri, misalkan tipe data titik yang terdiri dari koordinat x dan y. Bahasa Java telah menggunakan sebuah kelas untuk menyatakan tipe data titik karena bahasa pemrograman Java merupakan bahasa pemrograman berorientasi objek murni sehingga tidak mengenal struktur, melainkan mengenal apa yang disebut dengan kelas.

Perbedaan sebuah kelas dengan sebuah struktur adalah sebuah kelas dapat berdiri sendiri dan dapat digunakan untuk berbagai keperluan kelas-kelas yang lain, sedangkan sebuah struktur tidak dapat berdiri sendiri. Sebuah kelas lebih fleksibel untuk digunakan oleh kelas lain tanpa harus membongkar kode program utama, sedangkan jika digunakan struktur maka kode program harus dibongkar untuk disalin bagian strukturnya ke kode program utama yang lain.

Sebuah file dapat terdiri dari berbagai kelas, namun biasanya pada bahasa pemrograman Java sebuah file hanya terdiri dari satu kelas yang disimpan dengan nama kelas, misal file List.java berisi kelas List. Namun jika kelas yang dibuat misalnya public class nama\_kelas, maka kelas itu harus disimpan dalam satu file hanya untuk satu kelas. Setelah dilakukan kompilasi maka pada Java akan ada sebuah file .class yang berisi bytecode dari setiap kelas.

Jika sebuah file terdiri dari dua kelas, maka setelah dilakukan kompilasi akan dihasilkan dua buah file .class yang nantinya akan dibaca oleh interpreter Java saat program dieksekusi. Sebuah kelas saat program dieksekusi dan perintah new dijalankan, maka akan dibuat sebuah objek. Class adalah template untuk pembuatan objek.

Class juga memiliki anggota, diantaranya : atribut dan method.

Pemahaman singkatnya :

- Class merupakan pola / template yang menggambarkan kumpulan object yang mempunyai sifat dan perilaku yang sama.
- Class merupakan struktur data dari object (blueprint dari sebuah object), dan proses pembuatan object dari sebuah class disebut juga dengan instantiation.
- Class merupakan definisi data dan fungsi dari suatu hal (benda, konsep, dll).
- Di dalam class dideklarasikan variabel dan method yang dimiliki oleh obyek

### ***Karakteristik Class***

Class memiliki beberapa karakteristik, diantaranya : anggota class terdiri dari atribut dan method. Tiap-tiap anggota class memiliki control pengaksesan tersendiri, maksudnya adalah apakah anggota class tersebut dapat diakses dengan bebas (dengan tipe public) atau hanya dapat diakses melalui sebuah interface.

Dalam hal ini, interface adalah device yang digunakan untuk komunikasi antar object berbeda yang tidak memiliki hubungan apapun. Interface bisa dikatakan sebagai protokol komunikasi antar object tersebut.

## **4. OBJECT**

Merupakan perwujudan dari class, setiap object akan mempunyai attribute dan method yang dimiliki oleh class-nya, contohnya: amir, ahmad, yani merupakan object dari class manusia. Setiap object dapat berinteraksi dengan object lainnya meskipun berasal dari class yang berbeda.

### ***Definisi dan Gambaran Umum Object***

Object merupakan sesuatu yang memiliki identitas (nama), pada umumnya juga memiliki data tentang dirinya maupun object lain dan mempunyai kemampuan untuk melakukan sesuatu dan bisa bekerja sama dengan objek lainnya. Pada dasarnya semua benda yang ada di dunia dapat dianggap sebagai sebuah object. Rumah, mobil, sepeda motor, meja, dan komputer merupakan contoh-contoh object yang ada di dunia nyata. Object adalah implementasi dari class. Secara sederhananya, dapat dikatakan terdiri dari properti (atribut) dan method.

Pemahaman singkatnya :

- Objek merupakan sebuah objek khusus yang digunakan untuk mengakses anggota objek (atribut atau method) dari dalam objek itu sendiri.
- Apabila terdapat variabel lain yang memiliki nama yang sama dengan nama atribut di dalam objek, "this" dapat digunakan untuk merujuk ke atribut objek.

## ***Karakteristik Object***

Setiap object memiliki dua karakteristik yang utama, yaitu atribut dan behavior. Atribut merupakan status object dan behavior merupakan tingkah laku dari object tersebut.

Contoh sederhana adalah sepeda. Object sepeda memiliki atribut : pedal, roda, dan gigi, kecepatan serta memiliki tingkah laku seperti pacu dan rem. Sama seperti penggambaran di dunia nyata, pengembangan program dengan konsep OOP juga menggunakan model object yang akan menyimpan atributnya dalam variabel dan tingkah lakunya dalam method atau fungsi-fungsi/prosedur.

Deskripsinya :

Contoh objek : Sepeda

- Sepeda memiliki atribut ( state ) : pedal, roda, jeruji, dan warna.
- Sepeda memiliki tingkah laku ( behaviour ) : kecepatannya menaik, kecepatannya menurun, dan perpindahan gigi sepeda.

Dalam pengembangan perangkat lunak berorientasi objek, objek dalam perangkat lunak akan menyimpan state-nya dalam variabel dan menyimpan informasi tingkah laku ( behaviour ) dalam method-method atau fungsi-fungsi/prosedur.

## **KONSEP-KONSEP DALAM OOP/PBO**

### **1. Abstraction**

Adalah suatu cara untuk melihat suatu object dalam bentuk yang lebih sederhana. Sebagai contoh kita tidak melihat sebuah mobil sebagai ribuan komponen elektronik, sistem mekanik dengan empat buah ban, jok, stir kemudi dan sebagainya. Dengan Abstraction, suatu sistem yang kompleks dapat dipandang sebagai kumpulan subsistem-subsistem yang lebih sederhana, seperti halnya mobil merupakan suatu sistem yang terdiri atas berbagai subsistem, seperti subsistem kemudi, subsistem pengereman dan sebagainya.

Abstraksi merupakan kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari “pelaku” abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan.

Proses, fungsi atau metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

### **2. Encapsulation**

Merupakan suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan interferensi atau penyalahgunaan dari luar sistem dan sekaligus menyederhanakan penggunaan sistem tersebut.

Memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak. Hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.

### **3. Inheritance**

Merupakan konsep mewariskan attribute dan method yang dimiliki oleh sebuah class kepada class turunannya.

Dengan konsep ini class yang dibuat cukup mendefinisikan attribute dan method yang spesifik didalamnya, sedangkan attribute dan method yang lebih umum akan didapatkan dari class yang menjadi induknya.

Mengatur polimorfisme dan enkapsulasi dengan mengijinkan objek didefinisikan dan diciptakan dengan jenis khusus dari objek yang sudah ada. Objek-objek ini dapat membagi (dan memperluas) perilaku mereka tanpa harus mengimplementasi ulang perilaku tersebut (bahasa berbasis-objek tidak selalu memiliki inheritas.)

#### **4. Polymorphism**

Merupakan konsep yang memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu object agar melakukan suatu tindakan yang mungkin secara prinsip sama tetapi secara proses berbeda.

Model data berorientasi objek disebut dapat memberi fleksibilitas yang lebih, kemudahan mengubah program dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya selain itu pendekatan OOP lebih mudah dikembangkan dan dirawat.

Polimorfisme melalui pengiriman pesan. Tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan, dengan metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesan tersebut dikirim.

Contohnya, bila sebuah burung menerima pesan “gerak cepat”, dia akan menggerakkan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tunggal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama. Hal ini berlawanan dengan bahasa fungsional yang mencapai polimorfisme melalui penggunaan fungsi kelas-pertama.

Semoga bermanfaat. ( - dari berbagai sumber - )

---

All is about imagination - Endang Kurniawan

Sumber : <https://endangkurniawan.com/article-pemrograman-berorientasi-objek-oop.html>